

3.5)消费端的自定义消费监听

一:从3.1-3.4)我们消费端来消费消息的方式都是通过如下方式消费消息

```
while (true) {  
    QueueingConsumer.Delivery delivery = queueingConsumer.nextDelivery();  
    System.out.println(new String(delivery.getBody()));  
}
```

我们需要一个更加优雅的方式来消费消息,通过继承DefaultConsumer类来实现自定义的消息端

代码演示: 自定义消费端

```
/**  
 * Created by Administrator on 2018/10/19.  
 */  
public class AngleCustomConsumer extends DefaultConsumer {  
    /**  
     * Constructs a new instance and records its association to the passed-in channel.  
     */  
    * @param channel the channel to which this consumer is attached  
    */  
    public AngleCustomConsumer(Channel channel) {  
        super(channel);  
    }  
    /**  
     * 处理消息  
     * @param consumerTag  
     * @param envelope  
     * @param properties  
     * @param body  
     * @throws IOException  
     */  
    public void handleDelivery(String consumerTag,Envelope envelope,AMQP.BasicProperties properties, byte[] body)  
        throws IOException  
    {  
        System.out.println("自定义的消息消费端");  
        System.out.println("consumerTag="+consumerTag);  
        System.out.println("envelope="+envelope);  
        System.out.println("properties="+properties);  
        System.out.println("body="+new String(body));  
    }  
}
```

消费者:

```
package com.hnnd.mq.CustomConsumer;  
import com.rabbitmq.client.Channel;
```

```

import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import java.io.IOException;
import java.util.concurrent.TimeoutException;
/**
 * Created by Administrator on 2018/10/19.
 */
public class Consumer {

    public static void main(String[] args) throws IOException, TimeoutException {
        ConnectionFactory connectionFactory = new ConnectionFactory();
        connectionFactory.setVirtualHost("/");
        connectionFactory.setHost("47.104.128.12");
        connectionFactory.setPort(5672);
        Connection connection = connectionFactory.newConnection();
        Channel channel = connection.createChannel();
        channel.exchangeDeclare("test.customconsumer.exchange", "direct", true, true, false, null);
        channel.queueDeclare("test.customconsumer.queue", true, false, true, null);
        channel.queueBind("test.customconsumer.queue", "test.customconsumer.exchange", "test.customconsumer.key");
        //设置自定义消费端
        channel.basicConsume("test.customconsumer.queue", true, new AngleCustomConsumer(channel));
    }
}

```

消息生产者:

```

package com.hnnd.mq.CustomConsumer;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import java.io.IOException;
import java.util.concurrent.TimeoutException;
/**
 * Created by Administrator on 2018/10/19.
 */
public class producer {
    public static void main(String[] args) throws IOException, TimeoutException {
        ConnectionFactory connectionFactory = new ConnectionFactory();
        connectionFactory.setVirtualHost("/");
        connectionFactory.setHost("47.104.128.12");
        connectionFactory.setPort(5672);
        Connection connection = connectionFactory.newConnection();
        Channel channel = connection.createChannel();
        channel.basicPublish("test.customconsumer.exchange", "test.customconsumer.key", false, null, "自定义消费端消息".getBytes());
    }
}

```